# Domain-Shifting Algorithm: A New Domain-Decomposition Scheme for Molecular Dynamics Simulations on Parallel Computers

Chee Chin Liew,[1,*] Tamio Ikeshoji,[1] Norio Saito[1] and Hiroshi Inomata[2]

[1] *Tohoku National Industrial Research Institute Nigatake 4-2-1, Sendai 983-8551, Japan*
[2] *Department of Chemical Engineering, Tohoku University Aramaki-aza Aoba, Sendai 980-8579, Japan*

(Received November 5, 1999)

A domain is conventionally defined as a stationary sub-region of the simulated system in a domain-decomposition scheme for molecular dynamics (MD) simulations on parallel computers. We proposed an algorithm where all domains pre-assigned to processors are shifted to a particular direction, beyond the displacement of particles in the system during a time-step or a period of small time-steps; as a result, it allows us to reduce the data transfer partners in the particle re-allocation procedure. We also proposed a systematic link-cell method that allows us to make use of small domain and reduces the amount of data to be transferred for updating the positions and forces of particles, in comparison to the conventional schemes. Benchmark studies of a three-dimensional Lennard-Jones system have been carried out using a parallel MD simulation program implemented via a MPI-based message-passing interface on several parallel computers. A result on a 16-CPU parallel computer system shows that the new scheme allows us to achieve a high parallel efficiency (over 75 %) for MD simulations of a system with relatively small number of particles per processor ($N/P < 500$).

## §1. Introduction

Various domain-decomposition (DD; or so-called spatial-decomposition) schemes for molecular dynamics (MD) simulations on parallel computers were proposed mainly aiming at very large-scale simulations.[1)-5)] Namely, the system size is over $N = 10^6$ molecules, where the domain's size ($L_{\mathrm{domain}}$) is relatively much larger than the force cut-off range ($R_{\mathrm{cut}}$). Steve Plimpton proposed a DD algorithm ($S1$ of Ref. 6)) allowing the use of small domain, where $L_{\mathrm{domain}} < R_{\mathrm{cut}}$, but did not devise a scheme to avoid redundant of force calculation ($S2$ of Ref. 6)). A parallel MD simulation loop via DD approaches, implemented with link-cell method to avoid redundant force calculation, requires procedures to

1. apply boundary condition or some other initialization tasks
2. copy coordinates data ($R[\,]$) to and from neighboring domains
3. construct list for link-cell method
4. calculate force ($F[\,]$)
5. copy $F[\,]$ data to and from neighboring domains
6. integrate the equation of motion
7. re-allocate particle (move $R[\,]$ and $V[\,]$) to and from neighboring domains

Among these, procedures 2, 5 and 7 are extra communication tasks on a parallel

[*] E-mail address: liew@tniri.go.jp

machine in comparison to a MD run on a serial machine. The "copy" tasks (in procedures 2 and 5) transfer data between processors responsible for a sub-region within the range of pair potential interaction. On the other hand, the "re-allocate" task (in procedure 7) involve its close neighbors (8 or 26 neighbors in 2-D or 3-D decomposition) in order to transfer all data associated with an atom that has changed sub-region to a new sub-region.

In this work, we propose new methods for each of these tasks to reduce the number of partners and amount of data transfer over the inter-processor communications. The low communication overhead of the new schemes should allow us to achieve a high parallel efficiency for MD simulations of a system with relatively small number of particles per processor.

## §2. New algorithms

### 2.1. *Compact link-cell method*

The number of partners and amount of data to be transferred in the "copy" tasks are mainly determined by the way we construct the link-cell for force calculation. Making use of the Newton's third law, we may construct a link-cell map which include half of those cells within the interaction range ($R_{\text{cut}}$), as shown in Fig. 1. The pair potential interactions among particles in a center cell with its neighboring cells can be schematically represented by cell-cell pointers as shown in type A of Fig. 2. In the case where $R_{\text{cut}}$ is shorter than the domain length ($L_{\text{domain}}$), as shown in Fig. 3, we would need to transfer $R[\,]$ data from 5 domains for 2-D decomposition or 17 domains for 3-D decomposition.

We propose a new method, referred to as "compact link-cell method", which makes use of complement link-cell maps. For a 2-D system with a 2-D decomposition we will have to use two complement link-cell maps, namely type B and type B*, which are results of decomposing the link-cell map of type A, as shown in Fig. 2. Note that the force calculation among particles in the local cell should be done only in type B, and should not be done in type B*. Replacing those cells of type A on the domain

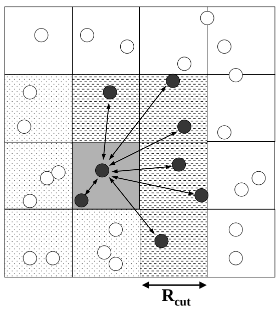

Fig. 1. 2-D representation of the link-cell method using the Newton's third law.

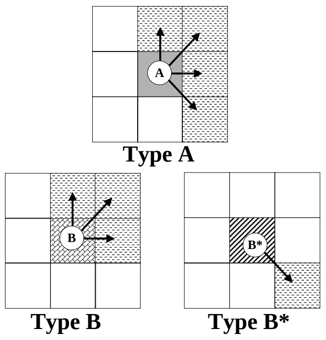

Fig. 2. Normal link-cell (type A) method and compact link-cell (type B and B*) method.

boundary with type B, and add the same amount of type B* to the another domain boundary, as shown in Fig. 4, we can reduce the number of data transfer partners from 5 to 3 domains on a 2-D decomposition. Extension of this systematic link-cell mapping method to a 3-D decomposition will allows us to reduce these from 17 to 7 domains.

David Brown et al. described a link-cell method[5] that appears to able to reduce the number of data transfer partners in the similar way as our method. However, their method consists of more complicated link-cell mapping types and, as have been mentioned by themselves in one of their papers,[7] it has a restriction that the domain

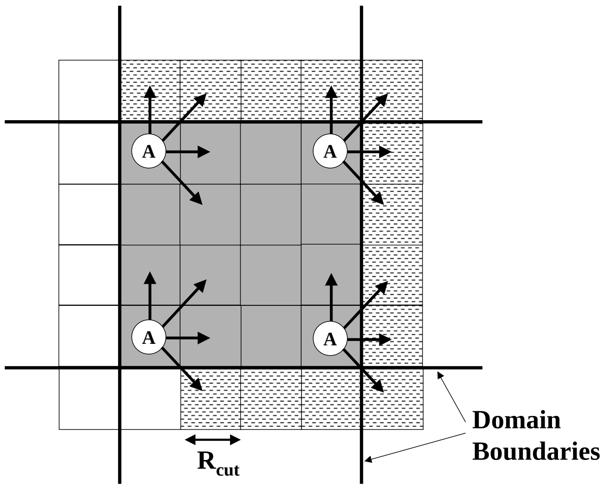

Fig. 3. Link-cell mapping using type A map on 2-D decomposition.

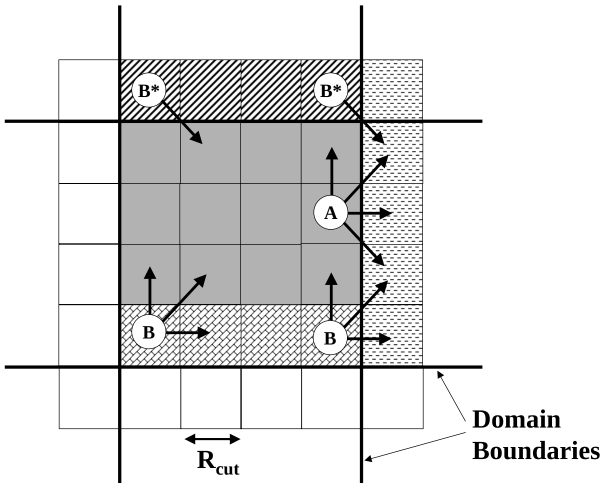

Fig. 4. Link-cell mapping using all types of maps on 2-D decomposition.

size must be larger than 2 $R_{\mathrm{cut}}$. On the other hand, our method allows us to make use of small domain, where the domain length is shorter than the interaction range ($L_{\mathrm{domain}} < R_{\mathrm{cut}}$). Figures 5 and 6 show an example where $R_{\mathrm{cut}} = 2\ L_{\mathrm{domain}}$ on a 2-D decomposition. Instead of using the type A which involve 12 domains, as in Fig. 5, we can construct link-cell maps using type B and B* which involve only 8 domains in the data transfer task, as illustrated in Fig. 6. Extension of this to a 3-D decomposition will allow us to reduce considerably the number of partners and amount of data to be transferred.

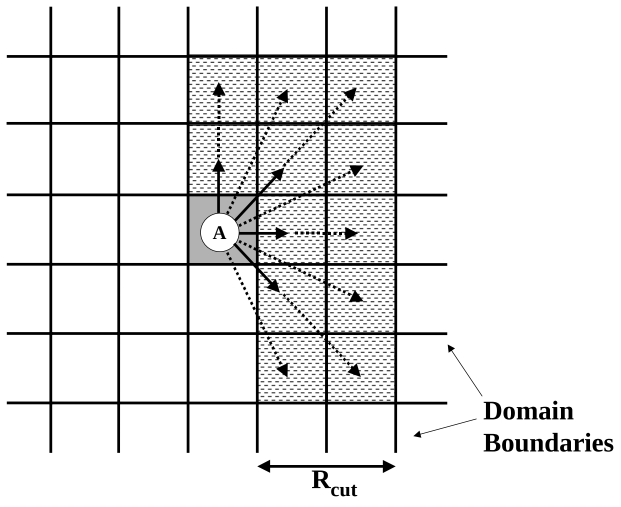

Fig. 5. Link-cell mapping using type A map when $R_{\mathrm{cut}} = 2L_{\mathrm{domain}}$.

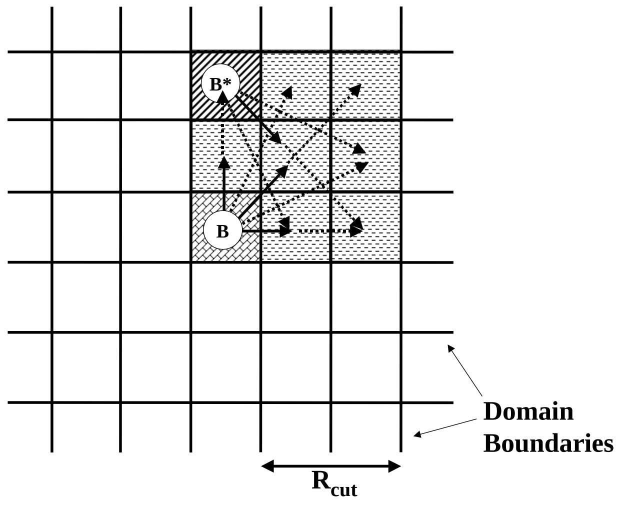

Fig. 6. Link-cell mapping using type B and B* when $R_{\mathrm{cut}} = 2L_{\mathrm{domain}}$.

2.2. *Shifting-domain method*

Conventionally, a domain in the domain-decomposition algorithm is defined as a stationary sub-region of the simulated system, which was pre-assigned to a processor that will be responsible on calculating force and up-dating coordinate of particles in the sub-region. After the integration of the equation of motion, for a time-step or a period of time-steps, some of the particles in a sub-region will move to a new sub-region, as shown in Fig. 7. A particle re-allocation procedure must be done, and it is obvious that this task will involve all neighbors of a domain to exchange all data associated with an atom that has moved to a new sub-region. Although the amount of the data to be transferred are small in comparison to those in the copy tasks, the number of partners of this communication task is larger, especially in the case of 3-D decomposition. This may become a very heavy task for a parallel computer having a date transfer device with a long latency time.

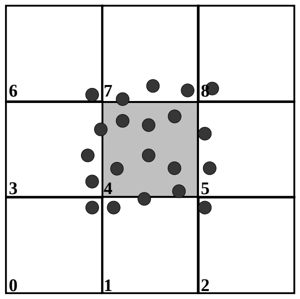

Fig. 7. Stationary domain of the conventional domain-decomposition schemes.

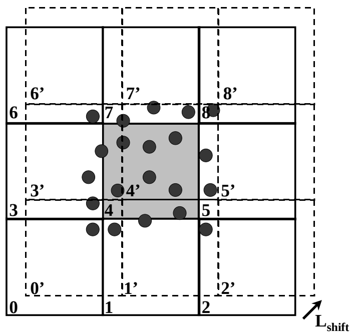

Fig. 8. Shifting domain of the new domain-decomposition scheme.

If we shift the sub-region towards a direction beyond the displacement of particles in the simulated system, as shown in Fig. 8, then we can see that the diffused particles will only move into 3 new sub-region (or domains) for a 2-D decomposition. Note that the magnitude of the domain shifting ($L_{\rm shift}$) should be as large as the so-called "skin" radius ($R_{\rm skin}$) used in the Verlet neighbor list method.[8] As a result, for the example shown in Fig. 8, the processor in the center which is responsible for the new sub-region (domain $4'$) now only need to send the data to 3 processors (domains $0'$, $1'$ and $3'$) and to receive the data of particles that move in from 3 processors on the other dicrections (domains $5'$, $7'$ and $8'$), instead of 8 domains each time. Although the domain-shifting do not reduce the amount of data to be transferred, it reduces the partners of data transfer and simplifies the data structure of the communication task for the parallel MD simulations, significantly in the case of 3-D decomposition.

## §3. Results and conclusions

Benchmark studies of three-dimensional Lennard-Jones system with 2-D decomposition have been carried out using a parallel MD simulation program implemented via a MPI-based message-passing interface on several parallel computers. We have counterchecked our simulation results with MD simulation program for serial machine. A test result of $N = 6912$, which is a relatively small number of particle even for a serial machine, on a network cluster (distributed-memory machines) containing 16 processors is listed in Table I. It shows that the low communication overhead of the new scheme allows us to achieve a high parallel efficiency (eff $> 75\%$) for MD simulations on a network cluster parallel machine even for system with relatively small number of particles per processor ($N/P < 500$), in comparison to eff $= 78\%$ on a Cray T3D (in Ref. 6)) for a higher number of particles per processor ($N/P \approx 2000$).

Table I. Results of MD runs using 3-dimensional LJ fluid at $T* = 0.72$, $\rho* = 0.8442$, $N = 6912$ via 2-D domian-decomposition algorithm with $R_{\rm cut} = 2.5\sigma, L_{\rm shift} = 0.5$ Å for 1000 time-steps on a RS/6000 8-nodes (2 CPU/node, Power-3 200MHz/CPU) network cluster machine connected by a Gigabit Ethernet switch. $t_{[P]}$ is the MD run time on P processor(s).

| P | $N/P$ | $t_{[P]}$[sec.] | speedup= $t_{[1]}/t_{[P]}$ | eff[%] |
|---|---|---|---|---|
| 1 | 6912 | 540.0 | 1 | 100 |
| 4 | 1728 | 142.5 | 3.718 | 93 |
| 9 | 768 | 78.97 | 6.938 | 75.98 |
| 16 | 432 | 44.72 | 11.88 | 75.5 |

We have proposed two algorithms for MD simulations on parallel computers. The domain-shifting algorithm, where the physical domains are shifted to a particular direction beyond the displacement of particles in the simulation system during a time-step or a period of small time-steps, reduces the data transfer partners in the particle re-allocation procedure. Combination of the domain-shifting algorithm with a new link-cell mapping technique, also proposed in this work, makes it possible to reduce at least half of the data transfer partners and the amount of data to be tansferred, in comparison to the conventional DD schemes. Further benchmark tests for comparison to various algorithms are now in progress.

## Acknowledgements

C. C. Liew would like to thank Japan Science and Technology Corporation (JST) for research fellowship. Benchmark test runs of the algorithm were performed on RS/6000 cluster machine at Tohoku National Industrial Research Institute.

## References

1) D. Fincham, Mol. Simulation **1** (1987), 1.
2) D. C. Rapaport, Comput. Phys. Commun. **62** (1991), 217.
3) M. R. S. Pinches, D. J. Tildesley and W. Smith, Mol. Simulation **6** (1991), 51.
4) W. Smith, Comput. Phys. Commun. **62** (1991), 229.
5) D. Brown, J. H. R. Clarke, M. Okuda and T. Yamazaki, Comput. Phys. Commun. **74** (1993), 67.
6) S. Plimpton, J. Comput. Phys. **117** (1995), 1.
7) D. Brown, J. H. R. Clarke, M. Okuda and T. Yamazaki, Comput. Phys. Commun. **83** (1994), 1.
8) L. Verlet, Phys. Rev. **159** (1967), 98.